Data-Driven Control of Cellular Networks

Sandeep Chinchali, Marco Pavone, Sachin Katti
Stanford University
Data-Driven Network Control is Ubiquitous

Cell Congestion

Video Streaming

Robotic Taxi Fleets

IoT Sensor Updates

Optimal Control
Challenges of Network Control

1. Data-driven forecasts
   • What *features/statistics* are needed for control?

2. Many Input Variables
   • Forecaster and Controller

3. Increasingly:
   • Data boundaries
General Approach

Joint (Public) State

Forecasted Features (Control API)

Controller

Action

Forecaster

Controller Private State

Forecaster Private State
Video Streaming

Past Throughputs

Future Throughputs (Risk-adjusted, ~30s)

Forecaster

Controller

Video QoE

Network Operator

Cloud Video Services

Private: User mobility

Private: Buffer State

QoE = \sum_{k=0}^{K} \text{Quality(Bitrate)} - \sum_{k=0}^{K} \text{Stalls} - \sum_{k=1}^{K-1} |\text{Quality}_{k+1} - \text{Quality}_{k}|
Robotic Taxi Fleet

Network Operator

Forecaster

Future Cell Congestion, Anomalies (~ hrs)

Controller

Passenger Wait Time, Ride Efficiency

Private: User Location, Cell Demand

City-wide Congestion (Google Maps)

Taxi Operator

Private: Taxi locations, Outstanding Demand

Taxi Routes

City-wide wide Congestion
Approach: Reinforcement Learning (RL)
Reinforcement Learning (RL)

**Goal:** Maximize the total reward

\[ \sum_t r_t \]

Adapted from Pensieve (Sigcomm 18, Mao et. al.)
Why is IoT traffic scheduling hard?

Contending goals
- Max IoT data
- Loss to mobile traffic
- Network limits

Optimal Control
RL Schedules Sensor Updates

1. Network State Space (Cell congestion forecasts)
2. IoT Scheduler Actions (Traffic Rate)
3. Operator policies/reward: efficient use of network
RL Dynamics: Live Network Experiments

\[ p(s_{t+1} \mid s_t, a_t) \]

\[ C_{t+1} = \begin{cases} 
C_t + Ma_t + \Delta \tilde{C}_t + \epsilon_t & \text{if } a_t > 0 \\
\tilde{C}_t + \Delta \tilde{C}_t + \epsilon_t & \text{if } a_t = 0 
\end{cases} \]
RL exploits transient dips in utilization

Controlled Congestion

Utilization gain

Transient Dip
Application 2: Mobile Video Streaming

How will forecasts of network conditions improve ABR?

Figure from Pensieve (Sigcomm 18, Mao et. al.)
Palo Alto Cell Throughput Diversity

**Insight:** Foresight of *true* network condition helps

**Solution:** Dynamically splice specialized controllers (metaRL)
High-Level Trace Statistics $[\mu, \sigma]$ (API)

Private: User Location

Infer network condition from forecast stats

QoE

Bitrate
Palo Alto (Our data) + FCC/Norway (Pensieve)
Generalize to FCC/Norway data from Pensieve

![Graph showing metaRL, train_test: test]
Re-analysis of Pensieve (Sigcomm 18, Mao et. al.)

\[
QoE = \sum_{k=0}^{K} \text{Quality(Bitrate)} - \sum_{k=0}^{K} \text{Stalls} - \sum_{k=1}^{K-1} |\text{Quality}_{k+1} - \text{Quality}_k|
\]

Linear QoE (hi-thpt)

HD QoE (vlo-thpt)

Optimize Tail
Future work

1. Broad-vision for Time-Series Control
   • Data-driven forecasts/ control strategies
     • *Intrinsic data boundaries*

2. Value/Price of Information used for Long-Term Control?

3. Privacy/Information Leakage

Questions: csandeep@stanford.edu
Claim: **Decouple** but *co-design* predictor and controller

Why not *end-to-end* learning?

Why **Decouple**?
1. Natural Data Boundaries
2. Modularity (Re-use forecaster)

Why **Co-design**?
1. Tune forecasts to control risk
2. Robust Adversarial Training
RL Formulation

\( \mathcal{M}^F = (S^F, A^F, \mathcal{T}^F, R^F) \)

\( \mathcal{M}^C = (S^C, A^C, \mathcal{T}^C, R^C) \)

\( s^F_t = \begin{bmatrix} x^F_{t,p} \\ x^F_t \\ x^J_t \end{bmatrix} \)

\( r^F_t = -r^C_t \)

\( a^F_t = \phi(s^F_t) \)

\( a^C_t \)

\( s^C_t = \begin{bmatrix} x^C_{t,p} \\ x^J_t \\ \phi(s^F_t) \end{bmatrix} \)

Adversarial
Quantifying Sub-optimality Gap

With oracle knowledge of network condition

Have to learn network condition

Value/Price of Timeseries Variables?
IoT Traffic Scheduling (AAAI 2018)

Diverse city-wide cell patterns